GIAC Intrusion Detection Curriculum Practical Assignment

Mark Scott
Parliament Hill 2000

Assignment 1

Detect 1

Log 1

```
16:44:17.003212 hackerdude.com.3270 > mynetwork.com.20: udp 9 (ttl 114, id 17849)
16:44:17.004646 hackerdude.com.3271 > mynetwork.com.9: udp 9 (ttl 114, id 18105)
16:44:17.009443 hackerdude.com.3272 > mynetwork.com.9: udp 9 (ttl 114, id 18361)
16:44:17.010702 hackerdude.com.3273 > mynetwork.com.514: udp 9 (ttl 114, id 18617)

this entry appeared regularly in the log, always on dst port 161 which is the defined SNMP port
16:44:17.015558 hackerdude.com.3274 > mynetwork.com.161: |47|45[len7<asnlen69] (ttl 114, id 18873)
16:44:17.016962 hackerdude.com.3275 > mynetwork.com.23: udp 9 (ttl 114, id 19129)
16:44:17.092775 hackerdude.com.3276 > mynetwork.com.512: udp 9 (ttl 114, id 20153)
16:44:17.096561 hackerdude.com.3277 > mynetwork.com.19: udp 9 (ttl 114, id 20409)
16:44:17.098823 hackerdude.com.3278 > mynetwork.com.109: udp 9 (ttl 114, id 20665)
16:44:17.102846 hackerdude.com.3279 > mynetwork.com.110: udp 9 (ttl 114, id 20921)
16:44:17.105063 hackerdude.com.3280 > mynetwork.com.109: udp 9 (ttl 114, id 21177)
16:44:17.108968 hackerdude.com.3281 > mynetwork.com.109: udp 9 (ttl 114, id 21433)
16:44:17.169020 hackerdude.com.3282 > mynetwork.com.17: udp 9 (ttl 114, id 22201)
16:44:17.170004 hackerdude.com.3283 > mynetwork.com.80: udp 9 (ttl 114, id 22457)
16:44:17.175217 hackerdude.com.3284 > mynetwork.com.15: udp 9 (ttl 114, id 22713)
16:44:17.266382 hackerdude.com.3285 > mynetwork.com.21: udp 9 (ttl 114, id 22969)
16:44:17.267204 hackerdude.com.3286 > mynetwork.com.25: udp 9 (ttl 114, id 23225)
16:44:17.272162 hackerdude.com.3287 > mynetwork.com.512: udp 9 (ttl 114, id 23481)
16:44:17.273136 hackerdude.com.3288 > mynetwork.com.119: udp 9 (ttl 114, id 23737)
16:44:17.336605 hackerdude.com.3289 > mynetwork.com.512: udp 9 (ttl 114, id 23993)
16:44:17.338070 hackerdude.com.3290 > mynetwork.com.161: |47|45[len7<asnlen69] (ttl 114, id 24249)
16:44:17.345269 hackerdude.com.3291 > mynetwork.com.13: udp 9 (ttl 114, id 24505)
16:44:17.345795 hackerdude.com.3292 > mynetwork.com.21: udp 9 (ttl 114, id 24761)
16:44:17.349045 hackerdude.com.3293 > mynetwork.com.515: udp 9 (ttl 114, id 25017)
```

```
16:44:17.407309 hackerdude.com.3294 > mynetwork.com.13: udp 9 (ttl 114, id 25273)
16:44:17.409412 hackerdude.com.3295 > mynetwork.com.11: udp 9 (ttl 114, id 25529)
16:44:17.413878 hackerdude.com.3296 > mynetwork.com.20: udp 9 (ttl 114, id 25785)
16:44:17.416481 hackerdude.com.3297 > mynetwork.com.513: udp 9 (ttl 114, id 26041)
16:44:17.419931 hackerdude.com.3298 > mynetwork.com.17: udp 9 (ttl 114, id 26297)
```

- you would have expected hackerdude.com.3299 here but it was not in the log

```
16:44:17.506392 hackerdude.com.3300 > mynetwork.com.109: udp 9 (ttl 114, id 26809)
16:44:17.507086 hackerdude.com.3301 > mynetwork.com.20: udp 9 (ttl 114, id 27065)
16:44:17.512545 hackerdude.com.3302 > mynetwork.com.514: udp 9 (ttl 114, id 27321)
16:44:17.513263 hackerdude.com.3303 > mynetwork.com.109: udp 9 (ttl 114, id 27577)
16:44:17.518749 hackerdude.com.3304 > mynetwork.com.11: udp 9 (ttl 114, id 27833)
16:44:17.519443 hackerdude.com.3305 > mynetwork.com.20: udp 9 (ttl 114, id 28089)
16:44:17.618639 hackerdude.com.3306 > mynetwork.com.15: udp 9 (ttl 114, id 28345)
16:44:17.619331 hackerdude.com.3307 > mynetwork.com.512: udp 9 (ttl 114, id 28601)
16:44:17.624872 hackerdude.com.3308 > mynetwork.com.515: udp 9 (ttl 114, id 28857)
16:44:17.625566 hackerdude.com.3309 > mynetwork.com.11: udp 9 (ttl 114, id 29113)
16:44:17.631101 hackerdude.com.3310 > mynetwork.com.7: udp 9 (ttl 114, id 29369)
16:44:17.631819 hackerdude.com.3311 > mynetwork.com.110: udp 9 (ttl 114, id 29625)
16:44:17.738979 hackerdude.com.3312 > mynetwork.com.109: udp 9 (ttl 114, id 29881)
16:44:17.739907 hackerdude.com.3313 > mynetwork.com.11: udp 9 (ttl 114, id 30137)
16:44:17.745216 hackerdude.com.3314 > mynetwork.com.37: udp 9 (ttl 114, id 30393)
16:44:17.746110 hackerdude.com.3315 > mynetwork.com.513: udp 9 (ttl 114, id 30649)
16:44:17.751331 hackerdude.com.3316 > mynetwork.com.513: udp 9 (ttl 114, id 30905)
16:44:17.752361 hackerdude.com.3317 > mynetwork.com.513: udp 9 (ttl 114, id 31161)
```

Log 2

59	21:44:17	UDP port scan	hackerdude.com	mynetwork.com	port=3|7|9|11|13|15|17-20|23|25|37|80|104-105|109-110|119|262-263|512-515|762-763|920-921|1006-1007|1022-1023

1. **Source of trace:**

   This event was taken from my web server.

2. **Detect was generated by:**

   Log 1
   WinDump (fields defined below) which is the porting to the Windows platform of *tcpdump*.
   URL: http://netgroup-serv.polito.it/windump/

```
Time                Source                  Destination       Protocol  Packet size
16:44:17.619331 hackerdude.com.3307 > mynetwork.com.512:   udp      9   ttl 114, id 28601)
```

   Log 2
   BlackIce Defender (fields defined below)
   URL: http://www.networkice.com

Severity/GMT		Intrusion	SRC	DEST	Parameter
59	21:44:17	UDP port scan	hackerdude.com	mynetwork.com	port=3|7|9|11|13|15|17-20|23|25|37|80|104-105|109-110|119|262-263|512-515|762-763|920-921|1006-1007|1022-1023

3. **Probability that the source was spoofed:**

The source was not likely spoofed. The attacker is scanning for open UDP ports.

4. **Description of the attack:**

This is a random scan of UDP ports, probably a reconnaissance probe to identify open ports or Trojans that use UDP ports such as BackOrifice (UDP 31337). It is more unlikely to be a denial-of-service attack since the IPs do not appear to be spoofed. There are some CVE entries (http://cve.mitre.org) associated with UDP scan events.

Name	Description
CVE-1999-0063	Cisco IOS 12.0 and other versions can be crashed by malicious UDP packets to the syslog port.
CVE-1999-0103	Echo and chargen, or other combinations of UDP services, can be used in tandem to flood the server, a.k.a. UDP bomb or UDP packet storm.
CVE-1999-0189	Solaris rpcbind listens on a high numbered UDP port, which may not be filtered since the standard port number is 111.
CVE-1999-0217	Malicious option settings in UDP packets could force a reboot in SunOS 4.1.3 systems.
CVE-1999-0378	InterScan VirusWall for Solaris doesn't scan files for viruses when a single HTTP request includes two GET commands.
CVE-1999-0438	Remote attackers can perform a denial of service in WebRamp systems by sending a malicious UDP packet to port 5353, changing its IP address.
CVE-1999-0514	UDP messages to broadcast addresses are allowed, allowing for a Fraggle attack that can cause a denial of service by flooding the target.
CVE-2000-0033	InterScan VirusWall SMTP scanner does not properly scan messages with malformed attachments.
CVE-2000-0221	The Nautica Marlin bridge allows remote attackers to cause a denial of service via a zero length UDP packet to the SNMP port.

5. **Attack Mechanism:**

This UDP port scan sends packets to random ports on the server. The *BlackIce* firewall blocked the scan so there was no response received by the attacker. UDP is a connection-less protocol that will respond in one or two ways. If the UDP port is open or blocked by a firewall there will be no response, but if the port is closed and not blocked by a firewall it will send an "ICMP Port Unreachable" message.

6. Correlations:

Many Trojans such as BackOrifice (UDP 31337), Deep Throat (UDP 2140), Donald Dick (UDP 23476), Trinoo (UDP 27444, 31335), Hack 'a' tack (UDP 31789, 31790), etc. listen on UDP ports and PCAnywhere listens on UDP 22 and 5632. So UDP port scanning is a good way for a hacker to find goldmines so to speak. Sans mentions in article http://www.sans.org/newlook/resources/IDFAQ/DT.htm about DeepThroat and http://www.sans.org/y2k/122399.htm shows some common UDP port sans.

7. Evidence of active targeting:

Yes there is evidence of active targeting. This server, a web server, was the only IP scanned out of a Class C address.

8. Severity:

Using the formula from class:

(Criticality + Lethal) – (System + Network Countermeasures) = Severity

so

(4+4) – (4+3)= 1

Criticality = 4 for this is a company web server
Lethal = 4 for if there had been a Trojan planted on this server or maybe a PCAnywhere compromise
System = 4 for this system has all patches and hotfixes available
Network = 3 for there is a firewall, but no IDS in place
Severity = 1

9. Defensive recommendations:

The defenses in place on this server were a firewall which blocks all ports except those ports associated with services that are allowed in this environment. This defense worked as the port scan was blocked by the firewall. It is recommended however that a dual IDS system would greatly enhance the perimeter protection by providing a snapshot of what is actually occurring at the perimeter entry and then again behind the firewall.

10. Multiple choice question:

Look at the following snippet from a WinDump (TCPDump for Windows) trace:

```
16:44:17.506392 hackerdude.com.3300 > mynetwork.com.109: udp 9 (ttl 114, id 26809)
16:44:17.507086 hackerdude.com.3301 > mynetwork.com.20: udp 9 (ttl 114, id 27065)
16:44:17.512545 hackerdude.com.3302 > mynetwork.com.514: udp 9 (ttl 114, id 27321)
16:44:17.513263 hackerdude.com.3303 > mynetwork.com.109: udp 9 (ttl 114, id 27577)
```

```
16:44:17.518749 hackerdude.com.3304 > mynetwork.com.11: udp 9 (ttl 114, id 27833)
16:44:17.519443 hackerdude.com.3305 > mynetwork.com.20: udp 9 (ttl 114, id 28089)
16:44:17.618639 hackerdude.com.3306 > mynetwork.com.15: udp 9 (ttl 114, id 28345)
```

Which of the following are true?

a.) this is a UDP port scan for the Deep Throat trojan
b.) this is a targeted attack
c.) the attack is spoofed
d.) it delivers a large payload
e.) both a and c

answer: b, because only one IP of a full Class C was scanned and this IP happen to be associated with the web server

---

Detect 2 – back to the top

Log 1 (the packets in this trace have been grouped together for clarity)

```
16:48:23.954196 imhacking.com.4079 > mynetwork.com.21: S 2495131:2495131(0) win 8192 ¬
 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id 42686)
16:48:23.954589 mynetwork.com.21 > imhacking.com.4079: S 29790329:29790329(0) ack 2495132 win 8760 ¬
 <mss 1460> (DF) (ttl 128, id 11978)
16:48:24.191755 imhacking.com.4079 > mynetwork.com.21: . ack 1 win 8760 (DF) (ttl 114, id 55998)
16:48:24.192706 imhacking.com.4079 > mynetwork.com.21: . ack 1 win 16384 (DF) (ttl 114, id 56254)
16:48:24.193570 mynetwork.com.21 > imhacking.com.4079: P 1:48(47) ack 1 win 8760 (DF) (ttl 128, id
13258)
16:48:24.361968 imhacking.com.4079 > mynetwork.com.21: P 1:17(16) ack 48 win 16337 (DF) (ttl 114, id
58302)
16:48:24.363225 mynetwork.com.21 > imhacking.com.4079: P 48:86(38) ack 17 win 8744 (DF) (ttl 128, id
14282)
16:48:24.556316 imhacking.com.4079 > mynetwork.com.21: P 17:33(16) ack 86 win 16299 (DF) (ttl 114, id
59326)
16:48:24.569618 mynetwork.com.21 > imhacking.com.4079: P 86:121(35) ack 33 win 8728 (DF) (ttl 128, id
15306)
16:48:24.763959 imhacking.com.4079 > mynetwork.com.21: R 2495164:2495164(0) win 0 (DF) (ttl 114, id
60606)
16:48:24.005405 imhacking.com.4086 > 2.mynetwork.com.21: S 2495132:2495132(0) win 8192  ¬
<mss 1460,nop,nop,sackOK> (DF) (ttl 114, id 44478)
16:48:24.005684 2.mynetwork.com.21 > imhacking.com.4086: S 29790340:29790340(0) ack 2495133 win 8760
¬
<mss 1460> (DF) (ttl 128, id 12234)
16:48:24.204336 imhacking.com.4086 > 2.mynetwork.com.21: . ack 1 win 8760 (DF) (ttl 114, id 57022)
16:48:24.205282 imhacking.com.4086 > 2.mynetwork.com.21: . ack 1 win 16384 (DF) (ttl 114, id 57278)
16:48:24.273846 2.mynetwork.com.21 > imhacking.com.4086: P 1:48(47) ack 1 win 8760 (DF) (ttl 128, id
13514)
16:48:24.457870 imhacking.com.4086 > 2.mynetwork.com.21: P 1:17(16) ack 48 win 16337 (DF) (ttl 114, id
58558)
16:48:24.461470 2.mynetwork.com.21 > imhacking.com.4086: P 48:86(38) ack 17 win 8744 (DF) (ttl 128, id
14794)
16:48:24.711499 imhacking.com.4086 > 2.mynetwork.com.21: P 17:33(16) ack 86 win 16299 (DF) (ttl 114,
id 60094)
16:48:24.712437 2.mynetwork.com.21 > imhacking.com.4086: P 86:121(35) ack 33 win 8728 (DF) (ttl 128,
id 16586)
```

```
16:48:24.954572 imhacking.com.4086 > 2.mynetwork.com.21: R 2495165:2495165(0) win 0 (DF) (ttl 114, id
62142)
16:48:24.006459 imhacking.com.4087 > 3.mynetwork.com.21: S 2495133:2495133(0) win 8192 ¬
<mss 1460,nop,nop,sackOK> (DF) (ttl 114, id 44734)
16:48:24.006695 3.mynetwork.com.21 > imhacking.com.4087: S 29790352:29790352(0) ack 2495134 win 8760
¬
<mss 1460> (DF) (ttl 128, id 12490)
16:48:24.197868 imhacking.com.4087 > 3.mynetwork.com.21: . ack 1 win 8760 (DF) (ttl 114, id 56510)
16:48:24.198778 imhacking.com.4087 > 3.mynetwork.com.21: . ack 1 win 16384 (DF) (ttl 114, id 56766)
16:48:24.280127 3.mynetwork.com.21 > imhacking.com.4087: P 1:48(47) ack 1 win 8760 (DF) (ttl 128, id
13770)
16:48:24.459228 imhacking.com.4087 > 3.mynetwork.com.21: P 1:17(16) ack 48 win 16337 (DF) (ttl 114, id
58814)
16:48:24.459950 3.mynetwork.com.21 > imhacking.com.4087: P 48:86(38) ack 17 win 8744 (DF) (ttl 128, id
14538)
16:48:24.709594 imhacking.com.4087 > 3.mynetwork.com.21: P 17:33(16) ack 86 win 16299 (DF) (ttl 114,
id 59838)
16:48:24.710513 3.mynetwork.com.21 > imhacking.com.4087: P 86:121(35) ack 33 win 8728 (DF) (ttl 128,
id 16330)
16:48:24.945604 imhacking.com.4087 > 3.mynetwork.com.21: R 2495166:2495166(0) win 0 (DF) (ttl 114, id
61630)
```

Log 2

39	21:48:24	FTP port probe	imhacking.com	mynetwork.com	port=21
39	21:48:24	FTP Login failed	imhacking.com	mynetwork.com	count=5&victim=x.x.x.x&login=anonymous
39	21:48:24	FTP port probe	imhacking.com	2.mynetwork.com	port=21
39	21:48:24	FTP Login failed	imhacking.com	2.mynetwork.com	count=5&victim=y.y.y.y&login=anonymous
39	21:48:24	FTP port probe	imhacking.com	3.mynetwork.com	port=21
39	21:48:24	FTP Login failed	imhacking.com	3.mynetwork.com	count=5&victim=z.z.z.z&login=anonymous

1. Source of trace:

This event was taken from my web server.

2. Detect was generated by:

Log 1

WinDump (fields defined below) which is the porting to the Windows platform of *tcpdump*.
URL: http://netgroup-serv.polito.it/windump/

```
Time            Source             Destination   Port F  Synch numbers     Data  Window size
16:48:23.954196 imhacking.com.4079 > mynetwork.com.21: S 2495131:2495131(0) win 8192 ¬
 max seg sz     options    select ack  frag
<mss 1460,nop,nop,sackOK> (DF) (ttl 114, id 42686)
```

Log 2

BlackIce Defender (fields defined below)
URL: http://www.networkice.com

Severity/GMT		Intrusion	SRC	DEST	Parameter
39	21:48:24	FTP Login failed	hackerdude.com	mynetwork.com	count=5&victim=x.x.x.x&login=anonymous

3. Probability that the source was spoofed:

The source was not likely spoofed because there is a 3-way handshake. The attacker is probing for open FTP servers.

4. Description of the attack:

This is a FTP server probe trying to identify open FTP servers and as such would be a reconnaissance event. There are several CVE entries (http://cve.mitre.org) that could be associated with a FTP server probe event. They are:

Name	CVE-1999-0777
Description	IIS FTP servers may allow a remote attacker to read or delete files on the server, even if they have "No Access" permissions

Name	CVE-1999-0017
Description	FTP servers can allow an attacker to connect to arbitrary ports on machines other than the FTP client, aka FTP bounce.

Name	CVE-1999-0349
Description	A buffer overflow in the FTP list (ls) command in IIS allows remote attackers to conduct a denial of service and, in some cases, execute arbitrary commands.

5. Attack Mechanism:

This FTP server probe tries to login into FTP servers by scanning the Internet for these servers. Once a FTP server is located the probe tries an anonymous login. This exploit can include DoS, access to private files, or hijacking a FTP server to serve illegal files. The *BlackIce* firewall blocked the attempted probe and login. This is evident because there is never a connection to port 20, the FTP-data port.

6. Correlations:

FTP is one of the oldest protocols used on the Internet and because of that there are many FTP exploitations. Most web publishing servers on the Internet also have a FTP server. Sans mentions in articles http://www.sans.org/y2k/091000.htm, http://www.sans.org/y2k/0105stutzman.htm, http://www.sans.org/y2k/word0105.htm, http://www.sans.org/y2k/083000.htm, http://www.sans.org/y2k/092200.htm, etc., about some FTP scans.

7. Evidence of active targeting:

No there is no evidence of active targeting. Every IP on this server was scanned.

8. Severity:

Using the formula from class:

(Criticality + Lethal) – (System + Network Countermeasures) = Severity

so

(4+4) – (4+3)= 1

Criticality = 4 for this is a company web server
Lethal = 4 for if there had been a login the server would have been comprised by possibly a DOS,
a system crash, files captured, etc.
System = 4 for this system has all patches and hotfixes available
Network = 3 for there is a firewall, but no IDS in place
Severity = 1

9. Defensive recommendations:

The defenses in place on this server were a firewall which blocks anonymous FTP logins. The FTP server was also configured to not allow anonymous logins. This defense worked as the FTP probe was blocked. It is recommended however that a dual IDS system would greatly enhance the perimeter protection by providing a snapshot of what is actually occurring at the perimeter entry and then again behind the firewall. This method would allow one to see what packets actually were blocked.

10. Multiple choice question:

Look at the following snippet from a WinDump (TCPDump for Windows) trace:

```
16:48:24.006459 imhacking.com.4087 > 3.mynetwork.com.21: S 2495133:2495133(0) win 8192 ¬
<mss 1460,nop,nop,sackOK> (DF) (ttl 114, id 44734)
16:48:24.006695 3.mynetwork.com.21 > imhacking.com.4087: S 29790352:29790352(0) ack 2495134 win 8760
¬
<mss 1460> (DF) (ttl 128, id 12490)
16:48:24.197868 imhacking.com.4087 > 3.mynetwork.com.21: . ack 1 win 8760 (DF) (ttl 114, id 56510)
16:48:24.198778 imhacking.com.4087 > 3.mynetwork.com.21: . ack 1 win 16384 (DF) (ttl 114, id 56766)
16:48:24.280127 3.mynetwork.com.21 > imhacking.com.4087: P 1:48(47) ack 1 win 8760 (DF) (ttl 128, id
13770)
16:48:24.459228 imhacking.com.4087 > 3.mynetwork.com.21: P 1:17(16) ack 48 win 16337 (DF) (ttl 114, id
58814)
16:48:24.459950 3.mynetwork.com.21 > imhacking.com.4087: P 48:86(38) ack 17 win 8744 (DF) (ttl 128, id
14538)
16:48:24.709594 imhacking.com.4087 > 3.mynetwork.com.21: P 17:33(16) ack 86 win 16299 (DF) (ttl 114,
id 59838)
```

```
16:48:24.710513 3.mynetwork.com.21 > imhacking.com.4087: P 86:121(35) ack 33 win 8728 (DF) (ttl 128,
id 16330)
16:48:24.945604 imhacking.com.4087 > 3.mynetwork.com.21: R 2495166:2495166(0) win 0 (DF) (ttl 114, id
61630)
```

Which of the following are true?

a.) this is a FTP server probe with a successful login
b.) this is a FTP server probe with a failed login
c.) this attack is a ICQTrojan using port 4087
d.) none of the above

answer: b, because there was never a FTP-data (port 20) connection

---

Detect 3 – back to the top

Log 1 (the packets in this trace have been grouped together for clarity)

```
16:46:09.304092 netbiosprobe.com.3318 > x.x.x.0.139: S 2360596:2360596(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id
29883)
16:46:10.856131 netbiosprobe.com.3318 > x.x.x.0.139: S 2360596:2360596(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id
39611)
16:46:12.923754 netbiosprobe.com.3318 > x.x.x.0.139: S 2360596:2360596(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id
48315)
16:46:13.502860 netbiosprobe.com.3318 >x.x.x.0.139: S 2360596:2360596(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id
58043)
16:46:09.301160 netbiosprobe.com.3320 > x.x.x.2.139: S 2360596:2360596(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id
30395)
16:46:12.359939 netbiosprobe.com.3320 > x.x.x.2.139: S 2360596:2360596(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id
43451)
16:46:30.603019 netbiosprobe.com.3320 >x.x.x.2.139: S 2360596:2360596(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id
38844)
16:46:09.311435 netbiosprobe.com.3321 > x.x.x.3.139: S 2360596:2360596(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id
30651)
16:46:09.314693 x.x.x.3.139 > netbiosprobe.com.3321: S 2010132586:2010132586(0) ack 2360597 win 8760 <mss 1460> (DF) (ttl 128,
id 50725)
16:46:10.661895 netbiosprobe.com.3321 > x.x.x.3.139: . ack 1 win 8760 (DF) (ttl 114, id 38331)
16:46:13.311119 netbiosprobe.com.3321 > x.x.x.3.139: F 1:1(0) ack 1 win 8760 (DF) (ttl 114, id 49339)
16:46:13.312646 x.x.x.3.139 > netbiosprobe.com.3321: F 1:1(0) ack 2 win 8760 (DF) (ttl 128, id 50981)
16:46:13.490319 netbiosprobe.com.3321 > x.x.x.3.139: . ack 2 win 8760 (DF) (ttl 114, id 57531)
16:46:09.419544 netbiosprobe.com.3327 >x.x.x.20.139: S 2360597:2360597(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id
32187)
16:46:12.286719 netbiosprobe.com.3327 >x.x.x.20.139: S 2360597:2360597(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id
40379)
16:46:18.255969 netbiosprobe.com.3327 > x.x.x.20.139: S 2360597:2360597(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id
9148)
16:46:30.544679 netbiosprobe.com.3327 >x.x.x.20.139: S 2360597:2360597(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id
35772)
```

Log 2

39	21:46:09	NetBIOS port probe	netbiosprobe.com	x.x.x.y	port=139
39	21:46:12	NetBIOS port probe	netbiosprobe.com	x.x.x.z	port=139
39	21:46:13	NetBIOS port probe	netbiosprobe.com	x.x.x.x	port=139
39	21:46:13	NetBIOS port probe	netbiosprobe.com	x.x.x.x	port=139
39	21:46:18	NetBIOS port probe	netbiosprobe.com	x.x.x.x	port=139
39	21:46:30	NetBIOS port probe	netbiosprobe.com	x.x.x.x	port=139

1. **Source of trace:**

   This event was taken from my firewall server.

2. **Detect was generated by:**

   Log 1
   WinDump (fields defined below) which is the porting to the Windows platform of *tcpdump*.
   URL: http://netgroup-serv.polito.it/windump/

```
Time                   Source                Dest      Port  F    Synch numbers       Data  Window size max seg sz  options select ack      frag
16:46:09.304092 netbiosprobe.com.3318 > x.x.x.0.139: S 2360596:2360596(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id
29883)
```

   Log 2
   BlackIce Defender (fields defined below)
   URL: http://www.networkice.com

Severity/GMT		Intrusion	SRC	DEST	Parameter
39	21:46:09	NetBIOS port probe	netbiosprobe.com	x.x.x.y	port=139
39	21:46:12	NetBIOS port probe	netbiosprobe.com	x.x.x.z	port=139
39	21:46:13	NetBIOS port probe	netbiosprobe.com	x.x.x.x	port=139
39	21:46:13	NetBIOS port probe	netbiosprobe.com	x.x.x.x	port=139
39	21:46:18	NetBIOS port probe	netbiosprobe.com	x.x.x.x	port=139
39	21:46:30	NetBIOS port probe	netbiosprobe.com	x.x.x.x	port=139

3. **Probability that the source was spoofed:**

   The source was not likely spoofed because there is a 3-way handshake with x.x.x.3. The attacker is probing to see if port 139 is open and this also identifies the server as a Windows OS.

4. Description of the attack:

This is a Netbios probe trying to connect to port 139 of unprotected servers. There are several CVE entries (http://cve.mitre.org) that are associated with a Netbios probe event. They are:

Name	Description
CVE-1999-0153	Windows 95/NT out of band (OOB) data denial of service through NETBIOS port, aka WinNuke.
CVE-1999-0288	Denial of service in WINS with malformed data to port 137 (NETBIOS Name Service).
CVE-1999-0407	By default, IIS 4.0 has a virtual directory /IISADMPWD which contains files that can be used as proxies for brute force password attacks, or to identify valid users on the system.
CVE-1999-0810	Denial of service in Samba NETBIOS name service daemon (nmbd).
CAN-1999-0499	** CANDIDATE (under review) ** NETBIOS share information may be published through SNMP registry keys in NT.
CAN-1999-0518	** CANDIDATE (under review) ** A NETBIOS/SMB share password is guessable.
CAN-1999-0519	** CANDIDATE (under review) ** A NETBIOS/SMB share password is the default, null, or missing.
CAN-1999-0520	** CANDIDATE (under review) ** A system-critical NETBIOS/SMB share has inappropriate access control.
CAN-1999-0621	** CANDIDATE (under review) ** A component service related to NETBIOS is running.
CAN-2000-0347	** CANDIDATE (under review) ** Windows 95 and Windows 98 allow a remote attacker to cause a denial of service via a NetBIOS session request packet with a NULL source name.
CAN-2000-0673	** CANDIDATE (under review) ** The NetBIOS Name Server (NBNS) protocol does not perform authentication, which allows remote attackers to cause a denial of service by sending a spoofed Name Conflict or Name Release datagram, aka the "NetBIOS Name Server Protocol Spoofing" vulnerability.

5. Attack Mechanism:

This Netbios probe tries to connect to open Netbios ports by scanning the Internet for these servers. This is an exploit with known vulnerabilities as mentioned in the CVE references in step 4. These exploits can include DoS and access to file and printer shares. The *BlackIce* firewall blocked the attempted probe and all IPs except for one, x.x.x.3. The log demonstrates that x.x.x.3 allowed connection to port 139.

6. Correlations:

With more and more Windows OS servers on the Internet this exploit has become more common. This native Mircosoft can be accessible from the Internet and has been the source of some powerful exploits. Sans mentions
in articles http://www.sans.org/newlook/digests/SAC/windows.htm, http://www.sans.org/newlook/resources/IDFAQ/DIC.htm, http://www.infoworld.com/cgi-bin/displayArchive.pl?/99/04/o08-04.75.htm, http://www.sans.org/y2k/091600.htm, etc., about many Netbios exploits.

7. Evidence of active targeting:

No there is no evidence of active targeting. Every IP on this server was scanned.

8. Severity:

Using the formula from class:

(Criticality + Lethal) – (System + Network Countermeasures) = Severity

so

(4+5) – (3+1)= 5

Criticality = 4 for this is a company server
Lethal = 5 probe was able to connect to port 139
System = 3 for this system has all patches and hotfixes available
Network =1 for there is a firewall, but the server that was compromised was not behind it
Severity = 5

9. Defensive recommendations:

The defenses should include a firewall with all Netbios ports blocked from the external world, making sure that all service packs are at SP6a and an IDS should be in place to actually see what attempts are being made and what traffic is actually making through to the private network.

10. Multiple choice question:

Look at the following snippet from a WinDump (TCPDump for Windows) trace:

```
16:46:09.311435 netbiosprobe.com.3321 > x.x.x.3.139: S 2360596:2360596(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id
30651)
16:46:09.314693 x.x.x.3.139 > netbiosprobe.com.3321: S 2010132586:2010132586(0) ack 2360597 win 8760 <mss 1460> (DF) (ttl 128,
id 50725)
16:46:10.661895 netbiosprobe.com.3321 > x.x.x.3.139: . ack 1 win 8760 (DF) (ttl 114, id 38331)
16:46:13.311119 netbiosprobe.com.3321 > x.x.x.3.139: F 1:1(0) ack 1 win 8760 (DF) (ttl 114, id 49339)
16:46:13.312646 x.x.x.3.139 > netbiosprobe.com.3321: F 1:1(0) ack 2 win 8760 (DF) (ttl 128, id 50981)
16:46:13.490319 netbiosprobe.com.3321 > x.x.x.3.139: . ack 2 win 8760 (DF) (ttl 114, id 57531)
```

Which of the following are true?

a.) this is a classic WinNuke exploit
b.) this is a Netbios probe and it appears that there is a large data exchange
c.) Denial of service in WINS with malformed data to NETBIOS Name Service.
d.) none of the above

answer: d, this is just a Netbios port probe that found an open port 139, at this point there is no exploit evident

Detect 4 – back to the top

Log 1 (the packets in this trace have been grouped together for clarity)

```
14:17:14.825626 x.x.x.100.1592 > x.x.x.2.54320: S 6375945:6375945(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id 43330)
14:17:17.750729 x.x.x.100.1592 > x.x.x.2.54320: S 6375945:6375945(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id 44866)
14:17:23.749894 x.x.x.100.1592 > x.x.x.2.54320: S 6375945:6375945(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id 50498)
```

Log 2

59	19:17:36	TCP trojan horse probe	24.24.123.100	38.228.171.2	port=54320&name=Back_Orifice_2000

1. Source of trace:

This event was taken from my firewall server.

2. Detect was generated by:

Log 1
WinDump (fields defined below) which is the porting to the Windows platform of *tcpdump*.
URL: http://netgroup-serv.polito.it/windump/

```
      Time           Source    Port     Dest      Port   F   Synch numbers      Data  Window size max seg sz  options select ack    frag
14:17:14.825626 x.x.x.100.1592 > x.x.x.2.54320: S 6375945:6375945(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id 43330)
```

Log 2

BlackIce Defender (fields defined below)
URL: http://www.networkice.com

Severity/GMT		Intrusion	SRC	DEST	Parameter
59	19:17:36	TCP trojan horse probe	24.24.123.100	38.228.171.2	port=54320&name=Back_Orifice_2000

3. Probability that the source was spoofed:

The source was not likely spoofed. The exploiter is probing for the Back_Orifice_2000 Trojan (TCP 54320).

4. Description of the attack:

This is a Back_Orifice_2000 Trojan (TCP 54320) probe in which a Back_Orifice_2000 server is trying to connect to a client. There is a CVE entry (http://cve.mitre.org) that is associated with Back Orifice Trojan events. It is:

Name	Description
CAN-1999-0660	** CANDIDATE (under review) ** A hacker utility or Trojan Horse is installed on a system, e.g. NetBus, Back Orifice, Rootkit, etc.

5. Attack Mechanism:

The Back Orifice server tries to connect to an open Back Orifice client. This is an exploit with known vulnerabilities as mentioned in the CVE reference in step 4. This exploit allows complete control of the client. The *BlackIce* firewall blocked the attempted probe.

6. Correlations:

This well known Trojan is all over the Internet which makes this exploit a common one. Sans mentions in articles http://www.sans.org/infosecFAQ/back_orifice.htm, http://www.sans.org/newlook/resources/IDFAQ/trinoo.htm, http://www.bo2k.com/, http://www.norton.com/avcenter/venc/data/back.orifice.2000.trojan.html, http://vil.mcafee.com/dispVirus.asp?virus_k=10229&, etc., about Back Orifice exploits.

7. Evidence of active targeting:

Yes there is evidence of active targeting. The way the Back Orifice 2000 server works is that you must know the IP of the server you wish to configure.

8. Severity:

Using the formula from class:

(Criticality + Lethal) – (System + Network Countermeasures) = Severity

so

(4+4) – (4+3)= 1

Criticality = 4 for this is a company server
Lethal = 4
System = 4 for this system has all patches and hotfixes available
Network =3 for there is a firewall
Severity = 1

9. Defensive recommendations:

The defenses were ok here, the firewall blocked the probe. It would be good to run netstat –an and see if there is a Back_Orifice_2000 port listening. If you think you have been compromised then check the registry as follows:

Window 9X - \\HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUNSERVICES

Windows NT - \\HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\WINDOWS\CURRENT VERSION\RUN

10. Multiple choice question:

Look at the following snippet from a WinDump (TCPDump for Windows) trace:

```
14:17:14.825626 x.x.x.100.1592 > x.x.x.2.54320: S 6375945:6375945(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id 43330)
14:17:17.750729 x.x.x.100.1592 > x.x.x.2.54320: S 6375945:6375945(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id 44866)
14:17:23.749894 x.x.x.100.1592 > x.x.x.2.54320: S 6375945:6375945(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id 50498)
```

Which of the following are true?

a.) this is a UDP probe for Back Orifice
b.) this is a Back_Orifice_2000 server search
c.) this is a UDP scan for an unknown trojan
d.) none of the above

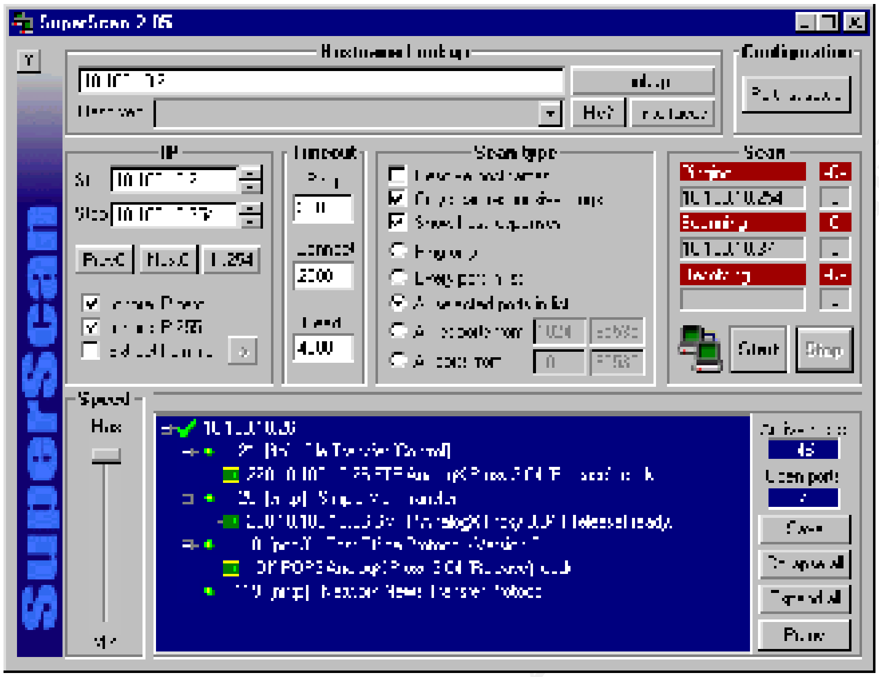
answer: b, this is a probe for Back_Orifice_2000 on TCP port 54320

Assignment 2 – back to the top

Evaluate an Attack

1. URL that the attack was acquired from

   http://www.members.home.com/rkeir/superscan.html, this is a powerful free TCP port scanner, pinger, and host resolver.



2. Describe the attack and how it works

   This attack is a reconnaissance exploit that will use SuperScan to remotely probe a Class C network in an attempt to find a
   vulnerable machine by determining its OS and monitoring which service ports are open. The first thing that must be done
   is to decide the ports that will be scanned. It is possible to scan the complete port range, but with a little thought here one
   can select certain ports that will most likely give the reconnaissance information desired. Many times the Operating System
   can be identified just by scanning the ports which many times will return a banner that IDs the OS.

   This network provided the following information from the port scan:
   -port 21: Microsoft FTP Service (version 4.0)
   -port 80: Microsoft-IIS/4.0
   -port 25: IMAIL (this is a mail server from http://www.ipswitch.com that runs on NT 4.0)
   -port 110: IMAIL (this is a mail server from http://www.ipswitch.com that runs on NT 4.0)
   -port 465: Microsoft SMTP Mail

   * + x.x.x.1

```
        |___   23  telnet
               |___ ..........(My.Network.com) Enter password: ....Incorrect password.....(My.Network.com)
Enter password:
   * + x.x.x.2
        |___   21  ftp
               |___ 220 smtp Microsoft FTP Service (Version 4.0)...
        |___   25  smtp
               |___ 220 X1 NT-ESMTP Server My.Network.com (IMail 5.05 2136-1)..
        |___   53  domain
        |___   80  http
               |___ HTTP/1.1 501 Not Supported..Server: Microsoft-IIS/4.0..Date: Sep 2000 16:33:05
               GMT..Content-Type:    text/html..Content-Le
        |___  106  3com-tsmux
               |___ 200 X1 NT-PWD Server My.Network.com (IMail 5.04)..
        |___  110  pop3
               |___ +OK X1 NT-POP3 Server My.Network.com (IMail 5.08 2785-1)..
        |___  135  epmap
        |___  443  https
        |___  465  ssmtp
               |___ 220- My.Network.com Microsoft SMTP MAIL ready at  Sep 2000 11:33:18 -0500
Version: 5.5.1877.197.19..220 ESMTP spoken he
   * + x.x.x.3
        |___   53  domain
        |___  135  epmap
```

It appears that this is a Microsoft Windows NT 4.0 server. If it had been Windows 2000 it would have been using IIS 5.0 instead of 4.0. Below is the port list used:

```
+,21,ftp,File Transfer [Control],C:\Program
Files\GlobalSCAPE\CuteFTP\cutftp32.exe,ftp://%a:%p/,\r\n
+,22,ssh,SSH Remote Login Protocol,,,
+,23,telnet,Telnet,telnet.exe,%a %p,\r\n
+,25,smtp,Simple Mail Transfer,,,
+,42,nameserver,WINS Host Name Server,,,
+,53,domain,Domain Name Server,,,
+,69,tftp,Trivial File Transfer,,,
+,79,finger,Finger,,,\r\n\r\n
+,80,http,World Wide Web HTTP,C:\Program Files\Internet
Explorer\IEXPLORE.EXE,http://%a:%p/,HEAD /\r\n\r\n
+,110,pop3,Post Office Protocol - Version 3,,,
+,111,sunrpc,SUN Remote Procedure Call,,,
+,119,nntp,Network News Transfer Protocol,,,
+,143,imap,Internet Message Access Protocol,,,
+,1080,socks,Socks,,,
+,1745,remote-winsock,remote-winsock,,,
+,2301,CIM,Compaq Insight Manager,,,
+,5190,aol,America-Online,,,
+,5191,aol-1,AmericaOnline1,,,
+,5192,aol-2,AmericaOnline2,,,
+,5193,aol-3,AmericaOnline3,,,
+,5631,pcanywheredata,pcANYWHEREdata,C:\Program Files\pcANYWHERE\Winaw32.exe,,
+,5632,pcanywherestat,pcANYWHEREstat,,,
+,5800,VNC,Virtual Network Computing server,C:\Program Files\ORL\VNC\vncviewer.exe,%a:0,
+,5900,VNC,Virtual Network Computing server,C:\Program Files\ORL\VNC\vncviewer.exe,%a:0,
+,6000,x11,-6063   X Window System,,,
+,8000,irdmi,iRDMI/Shoutcast Server,,,
+,8010,wingate-logfile,Wingate web logfile,,,
+,8080,WWW-Proxy,Standard HTTP Proxy,,,
+,9100,JetDirect,HP JetDirect Printer Server,,,
+,12345,Netbus,Win95/NT Netbus backdoor,,,
+,25867,WebCam32,WebCam32 Admin,,,
+,54320,BO2K,Back Orifice 2000,,,
```

Probing these ports will provide the type operating system and a list of vulnerabilities that can be used against it.

3. Provide an annotated network trace of the attack in action

This trace was captured using *WinDump* which is the porting to the Windows platform of *tcpdump*.
URL: http://netgroup-serv.polito.it/windump/

```
    Time            Source    Port    Dest  Port  F  Synch numbers    Data  Window size max seg sz  options select ack    frag
11:33:02.756725 x.x.x.100.2618 > x.x.x.2.1:  S 3013985:3013985(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id 9405)
```

This is just a snippet of a very large trace of a complete scan of a Class C address.

All ports are coded green

Open port 21 packets are coded yellow

Closed port 42 packets are coded gray

The next line begins the probe of port 21. Notice there is a Syn, Syn/Ack, and an Ack a few more lines below.

```
11:33:03.425908 y.y.y.100.2630 > x.x.x.2.21: S 3014479:3014479(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id 21181)
11:33:03.426136 x.x.x.2.21 > y.y.y.100.2630: S 11148055:11148055(0) ack 3014480 win 8760 <mss 1460> (DF) (ttl 128, id 19640)
11:33:03.455947 y.y.y.100.2631 > x.x.x.2.22: S 3014517:3014517(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id 23229)
11:33:03.456666 x.x.x.2.22 > y.y.y.100.2631: R 0:0(0) ack 3014518 win 0 (ttl 128, id 19896)
11:33:03.458396 y.y.y.100.2632 > x.x.x.2.23: S 3014534:3014534(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id 23485)
11:33:03.458488 x.x.x.2.23 > y.y.y.100.2632: R 0:0(0) ack 3014535 win 0 (ttl 128, id 20152)
11:33:03.494627 y.y.y.100.2634 > x.x.x.2.25: S 3014645:3014645(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id 25789)
11:33:03.494790 x.x.x.2.25 > y.y.y.100.2634: S 11148065:11148065(0) ack 3014646 win 8760 <mss 1460> (DF) (ttl 128, id 21432)
11:33:03.597290 y.y.y.100.2630 > x.x.x.2.21: . ack 1 win 8760 (DF) (ttl 114, id 29885)
```

The next few yellow lines show data are being sent here….. trying to log in as anonymous, then a FIN and Reset are shown.

```
11:33:03.598504 x.x.x.2.21 > y.y.y.100.2630: P 1:48(47) ack 1 win 8760 (DF) (ttl 128, id 23992)
11:33:03.599148 y.y.y.100.2630 > x.x.x.2.21: P 1:16(15) ack 1 win 8760 (DF) (ttl 114, id 30141)
11:33:03.641680 y.y.y.100.2634 > x.x.x.2.25: . ack 1 win 8760 (DF) (ttl 114, id 30653)
```

```
11:33:03.642553 y.y.y.100.2634 > x.x.x.2.25: P 1:2(1) ack 1 win 8760 (DF) (ttl 114, id 30909)
11:33:03.679560 x.x.x.2.25 > y.y.y.100.2634: P 1:57(56) ack 2 win 8759 (DF) (ttl 128, id 24504)
11:33:03.745359 x.x.x.2.21 > y.y.y.100.2630: . ack 16 win 8745 (DF) (ttl 128, id 24760)
11:33:03.965128 y.y.y.100.2634 > x.x.x.2.25: F 2:2(0) ack 57 win 8704 (DF) (ttl 114, id 35773)
11:33:03.965594 x.x.x.2.25 > y.y.y.100.2634: . ack 3 win 8759 (DF) (ttl 128, id 27064)
```

The next 2 lines are a probe of port 42. A Syn/Reset, no 3 way handshake. The scanner tries this closed port 2 more times.

```
11:33:03.966073 y.y.y.100.2644 > x.x.x.2.42: S 3015084:3015084(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id 36029)
11:33:03.966173 x.x.x.2.42 > y.y.y.100.2644: R 0:0(0) ack 3015085 win 0 (ttl 128, id 27320)
11:33:03.966714 x.x.x.2.25 > y.y.y.100.2634: F 57:57(0) ack 3 win 8759 (DF) (ttl 128, id 27576)
11:33:03.974365 y.y.y.100.2630 > x.x.x.2.21: F 16:16(0) ack 48 win 8713 (DF) (ttl 114, id 36541)
11:33:03.974755 x.x.x.2.21 > y.y.y.100.2630: P 48:93(45) ack 17 win 8745 (DF) (ttl 128, id 28088)
11:33:03.975229 x.x.x.2.21 > y.y.y.100.2630: F 93:93(0) ack 17 win 8745 (DF) (ttl 128, id 28344)
11:33:04.137390 y.y.y.100.2632 > x.x.x.2.23: S 3014534:3014534(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id 41661)
11:33:04.137484 x.x.x.2.23 > y.y.y.100.2632: R 0:0(0) ack 1 win 0 (ttl 128, id 31416)
11:33:04.146171 y.y.y.100.2634 > x.x.x.2.25: . ack 58 win 8704 (DF) (ttl 114, id 42685)
11:33:04.151619 y.y.y.100.2630 > x.x.x.2.21: R 3014496:3014496(0) win 0 (DF) (ttl 114, id 42941)
11:33:04.152323 y.y.y.100.2630 > x.x.x.2.21: R 3014496:3014496(0) win 0 (ttl 114, id 43197)
11:33:04.275046 y.y.y.100.2655 > x.x.x.2.53: S 3015525:3015525(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id 44221)
11:33:04.275227 x.x.x.2.53 > y.y.y.100.2655: S 11148071:11148071(0) ack 3015526 win 8760 <mss 1460> (DF) (ttl 128, id 32952)
11:33:04.426586 y.y.y.100.2655 > x.x.x.2.53: . ack 1 win 8760 (DF) (ttl 114, id 47549)
11:33:04.427438 y.y.y.100.2655 > x.x.x.2.53: P 1:2(1) ack 1 win 8760 (DF) (ttl 114, id 47805)
11:33:04.620280 y.y.y.100.2644 > x.x.x.2.42: S 3015084:3015084(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id 52669)
11:33:04.620385 x.x.x.2.53 > y.y.y.100.2655: . ack 2 win 8759 (DF) (ttl 128, id 39352)
11:33:04.635691 x.x.x.2.42 > y.y.y.100.2644: R 0:0(0) ack 1 win 0 (ttl 128, id 39608)
11:33:04.744683 y.y.y.100.2632 > x.x.x.2.23: S 3014534:3014534(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id 55997)
11:33:04.744960 x.x.x.2.23 > y.y.y.100.2632: R 0:0(0) ack 1 win 0 (ttl 128, id 42424)
11:33:04.822947 x.x.x.2.51 > y.y.y.100.2653: R 0:0(0) ack 1 win 0 (ttl 128, id 44472)
11:33:05.103962 y.y.y.100.2671 > x.x.x.2.69: S 3016350:3016350(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id 62653)
11:33:05.104971 x.x.x.2.69 > y.y.y.100.2671: R 0:0(0) ack 3016351 win 0 (ttl 128, id 49080)
11:33:05.332945 y.y.y.100.2632 > x.x.x.2.23: S 3014534:3014534(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id 3774)
11:33:05.333052 x.x.x.2.23 > y.y.y.100.2632: R 0:0(0) ack 1 win 0 (ttl 128, id 55736)
11:33:05.656202 y.y.y.100.2682 > x.x.x.2.80: S 3016899:3016899(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id 10942)
11:33:05.656453 x.x.x.2.80 > y.y.y.100.2682: S 11148082:11148082(0) ack 3016900 win 8760 <mss 1460> (DF) (ttl 128, id 62904)
11:33:05.734723 y.y.y.100.2671 > x.x.x.2.69: S 3016350:3016350(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id 11454)
11:33:05.734907 x.x.x.2.69 > y.y.y.100.2671: R 0:0(0) ack 1 win 0 (ttl 128, id 63416)
11:33:05.810363 y.y.y.100.2682 > x.x.x.2.80: . ack 1 win 8760 (DF) (ttl 114, id 13502)
11:33:05.811174 y.y.y.100.2682 > x.x.x.2.80: P 1:25(24) ack 1 win 8760 (DF) (ttl 114, id 13758)
11:33:05.814905 x.x.x.2.80 > y.y.y.100.2682: P 1:262(261) ack 25 win 8736 (DF) (ttl 128, id 65464)
11:33:05.816429 x.x.x.2.80 > y.y.y.100.2682: F 262:262(0) ack 25 win 8736 (DF) (ttl 128, id 185)
11:33:05.862996 y.y.y.100.2644 > x.x.x.2.42: S 3015084:3015084(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id 15806)
11:33:05.863088 x.x.x.2.42 > y.y.y.100.2644: R 0:0(0) ack 1 win 0 (ttl 128, id 2233)
11:33:06.002635 y.y.y.100.2682 > x.x.x.2.80: R 3016924:3016924(0) win 0 (DF) (ttl 114, id 18622)
11:33:06.003232 y.y.y.100.2682 > x.x.x.2.80: R 3016924:3016924(0) win 0 (ttl 114, id 18878)
```

Assignment 3 – back to the top

"Analyze this…"

Your organization has been asked to provide a bid to provide security services for this facility. You have been

allowed to run a Snort system with a fairly standard rulebase for a month. From time to time the power has failed,
or the disk was full so you do not have data for all days. Your task is to analyze the data, be especially alert for signs
of compromised systems or network problems and produce an analysis report.

Facility Analysis

As part of our bid to provide security services to your firm it is our intent to provide a *preliminary* snapshot of your network
traffic and identify any any network problems that may exist or any hostile traffic that may have compromised your environment.
As the traces from the Snort logs are analyzed it has become evident that seemingly hostile traffic has infiltrated your network.
We will describe our findings below, but we must mention at this point that a more thorough, daily around the clock scan be
conducted with a more refined rulebase set according to these preliminary findings.

The Log Analysis

All the traces below have been pulled from several logs that cover more than 30 days of scans
and is to be used as just an overview of the actual activity.

Wingate 1080 Attempt

```
07/17-21:04:23.836920  [**] WinGate 1080 Attempt [**] 208.240.218.220:4317 ->
MY.NET.97.31:1080
07/19-23:43:42.215212  [**] WinGate 1080 Attempt [**] 208.194.161.50:4855 ->
MY.NET.98.124:1080
07/26-00:43:48.107241  [**] WinGate 1080 Attempt [**] 64.86.6.70:2840 -> MY.NET.98.119:1080
07/28-23:41:41.050872  [**] WinGate 1080 Attempt [**] 206.50.68.20:1032 ->
MY.NET.98.177:1080
07/29-12:06:25.180739  [**] WinGate 1080 Attempt [**] 206.50.68.20:2737 ->
MY.NET.97.184:1080
```

There was an extremely large, almost daily amount of packets destined for a WinGate proxy server on this network. There are known exploits to the WinGate Proxy and keep in mind they are mainly used to forward attacks to other machines. This brings up an issue of liability. Because of this large number of probes it is possible or least it raises the question of whether or not there is a proxy running on your network. One should review all the internal IPs that the probes target and thoroughly investigate these machines.

Below are vulnerabilities listed at the CVE site. (http://cve.mitre.org)

Name	Description
CVE-1999-0290	The WinGate telnet proxy allows remote attackers to cause a denial of service via a large number of connections to localhost.
CVE-1999-0291	The WinGate proxy is installed without a password, which allows remote attackers to redirect connections without authentication.
CVE-1999-0441	Remote attackers can perform a denial of service in WinGate machines using a buffer overflow in the Winsock Redirector Service.
CVE-1999-0494	Denial of service in WinGate proxy through a buffer overflow in POP3.
CAN-1999-0657	** CANDIDATE (under review) ** WinGate is being used.

SYN-FIN scan!

```
06/28-06:52:51.003562  [**] SYN-FIN scan! [**] 202.0.178.98:53 -> MY.NET.1.135:53
07/19-09:49:19.680254  [**] SYN-FIN scan! [**] 212.171.169.46:15283 -> MY.NET.1.4:21
07/27-10:15:14.437118  [**] SYN-FIN scan! [**] 210.84.179.196:15396 -> MY.NET.60.8:113
07/29-15:30:46.393217  [**] SYN-FIN scan! [**] 212.177.241.139:80 -> MY.NET.1.5:80
08/01-00:04:42.262533  [**] SYN-FIN scan! [**] 207.0.62.254:1524 -> MY.NET.1.5:1524
08/03-19:59:23.475592  [**] SYN-FIN scan! [**] 149.225.111.69:7904 -> MY.NET.60.14:1302
```

This trace showed a large amount of SYN-FIN scans from a wide range of ports to a wide range of ports. This vulnerability is used many times trying to fingerprint an operating system. Never in normal traffic will a SYN and FIN flag be set.

Below are vulnerabilities listed at the CVE site. (http://cve.mitre.org)

Name	Description
CVE-1999-0116	Denial of service when an attacker sends many SYN packets to create multiple connections without ever sending an ACK to complete the connection, aka SYN flood.
CVE-1999-0770	Firewall-1 sets a long timeout for connections that begin with ACK or other packets except SYN, allowing an attacker to conduct a denial of service via a large number of connection attempts to unresponsive systems.
CAN-1999-0216	** CANDIDATE (under review) ** Denial of service of inetd on Linux through SYN and RST packets.
CAN-1999-0240	** CANDIDATE (under review) ** Some filters or firewalls allow fragmented SYN packets with IP reserved bits in violation of their implemented policy.

CAN-1999-0453	** CANDIDATE (under review) ** An attacker can identify a CISCO device by sending a SYN packet to port 1999, which is for the Cisco Dicsovery Protocol (CDP).
CAN-2000-0324	** CANDIDATE (under review) ** pcAnywhere 8.x and 9.x allows remote attackers to cause a denial of service via a TCP SYN scan, e.g. by nmap.

UDP Port Scan

```
Jul  8 15:13:23 24.3.0.33:53 -> MY.NET.97.49:1929 UDP
Jul  8 15:13:24 24.3.0.33:53 -> MY.NET.97.49:1930 UDP
Jul  8 15:13:24 24.3.0.33:53 -> MY.NET.97.49:1931 UDP
Jul  8 15:13:24 24.3.0.33:53 -> MY.NET.97.49:1932 UDP
Jul  8 15:13:24 24.3.0.33:53 -> MY.NET.97.49:1933 UDP
Jul  8 15:13:25 24.3.0.33:53 -> MY.NET.97.49:1934 UDP
```

This trace showed a complete UDP port scan on this machine my.net.87.49. UDP scans are used for mapping and for probing for Trojans.

Tiny Fragments – Possible Hostile Activity

```
07/11-03:33:54.281367  [**] Tiny Fragments - Possible Hostile Activity [**] 208.61.144.55 ->
MY.NET.230.241
07/26-11:05:01.522342  [**] Tiny Fragments - Possible Hostile Activity [**] 202.76.177.204 -
> MY.NET.70.20
```

This trace showed 'tiny fragments' which is not the usual in network traffic. This many times is used to bypass any detection. It is suggested that all targets in the the files be examined carefully. There are some that feel this exploit is used many times to install Trojans on unsuspected machines.

Below are other vulnerabilities listed at the CVE site. (http://cve.mitre.org)

Name	Description
CAN-2000-0133	** CANDIDATE (under review) ** Buffer overflows in Tiny FTPd 0.52 beta3 FTP server allows users to execute commands via the STOR, RNTO, MKD, XMKD, RMD, XRMD, APPE, SIZE, and RNFR commands.
CAN-2000-0630	** CANDIDATE (under review) ** IIS 4.0 and 5.0 allows remote attackers to obtain fragments of source code by appending a +.htr to the URL, a variant of the "File Fragment Reading via .HTR" vulnerability.

PortScan

```
07/17-21:21:28.390564  [**] spp_portscan: PORTSCAN DETECTED from 24.6.158.218 (STEALTH) [**]
07/17-21:21:29.634628  [**] spp_portscan: portscan status from 24.6.158.218: 1 connections
across 1 hosts: TCP(1), UDP(0) STEALTH [**]
```

```
07/17-21:21:31.427309  [**] spp_portscan: End of portscan from 24.6.158.218 (TOTAL HOSTS:1
TCP:1 UDP:0) [**]
```

This trace shows a PortScan. This occurred many times during the time the traces were collected. This usually signals a reconnaissance in in effect. This is the first sign to really begin watching the network closely. This information will probably be used in a future attack.

Queso fingerprint

```
07/17-21:11:17.806127  [**] Queso fingerprint [**] 192.203.80.142:3240 -> MY.NET.99.23:113
07/19-09:49:16.702569  [**] Queso fingerprint [**] 212.171.169.46:24122 -> MY.NET.1.3:21
07/19-09:49:22.702119  [**] Queso fingerprint [**] 212.171.169.46:22536 -> MY.NET.1.5:21
07/27-10:15:14.440976  [**] Queso fingerprint [**] 210.84.179.196:15398 -> MY.NET.60.8:113

07/12-12:46:34.921774  [**] Probable NMAP fingerprint attempt [**] 24.200.160.45:1548 ->
MY.NET.70.241:8899
```

Queso and namp are programs that are used to identify remote operating systems. This trace shows a queso and an nmap probe.

Below are vulnerabilities listed at the CVE site. (http://cve.mitre.org)

Name	Description
CAN-1999-0454	** CANDIDATE (under review) ** A remote attacker can sometimes identify the operating system of a host based on how it reacts to some IP or ICMP packets, using a tool such as nmap or queso.

Possible wu-ftpd exploit

```
07/19-03:53:00.191779  [**] site exec - Possible wu-ftpd exploit - GIAC000623 [**]
212.35.163.64:1245 -> MY.NET.100.165:21
07/29-12:07:56.525800  [**] site exec - Possible wu-ftpd exploit - GIAC000623 [**]
211.38.95.138:3048 -> MY.NET.156.127:21
```

This trace shows a possible attempt to exploit the wu-ftpd daemon.

Below are vulnerabilities listed at the CVE site. (http://cve.mitre.org)

Name	Description
CVE-1999-0075	PASV core dump in wu-ftpd daemon when attacker uses a QUOTE PASV command after specifying a username and password.
CVE-1999-0080	wu-ftp FTP server allows root access via "site exec" command.
CVE-1999-0081	wu-ftp allows files to be overwritten via the rnfr command.

CVE-1999-0368	Buffer overflows in wuarchive ftpd (wu-ftpd) and ProFTPD lead to remote root access, a.k.a. palmetto.
CVE-1999-0720	The pt_chown command in Linux allows local users to modify TTY terminal devices that belong to other users.
CVE-1999-0878	Buffer overflow in WU-FTPD and related FTP servers allows remote attackers to gain root privileges via MAPPING_CHDIR.
CVE-1999-0879	Buffer overflow in WU-FTPD and related FTP servers allows remote attackers to gain root privileges via macro variables in a message file.
CVE-1999-0880	Denial of service in WU-FTPD via the SITE NEWER command, which does not free memory properly.
CVE-1999-0955	Race condition in wu-ftpd and BSDI ftpd allows remote attackers gain root access via the SITE EXEC command.
CVE-1999-0997	wu-ftp with FTP conversion enabled allows an attacker to execute commands via a malformed file name that is interpreted as an argument to the program that does the conversion, e.g. tar or uncompress.
CAN-1999-0076	** CANDIDATE (under review) ** Buffer overflow in wu-ftp from PASV command causes a core dump.
CAN-1999-0156	** CANDIDATE (under review) ** wu-ftpd FTP daemon allows any user and password combination.
CAN-1999-0661	** CANDIDATE (under review) ** A system is running a version of software that was replaced with a Trojan Horse at its distribution point, e.g. TCP Wrappers, wuftpd, etc.
CAN-1999-0911	** CANDIDATE (under review) ** Buffer overflow in ProFTPD, wu-ftpd, and beroftpd allows remote attackers to gain root access via a series of MKD and CWD commands that create nested directories.
CAN-2000-0573	** CANDIDATE (under review) ** The lreply function in wu-ftpd 2.6.0 and earlier does not properly cleanse an untrusted format string, which allows remote attackers to execute arbitrary commands via the SITE EXEC command.

Watchlist

```
07/19-23:34:38.661309  [**] Watchlist 000222 NET-NCFC [**] 159.226.63.190:113 ->
MY.NET.253.41:55592
07/19-23:34:40.350907  [**] Watchlist 000222 NET-NCFC [**] 159.226.63.190:1963 ->
MY.NET.253.41:25
```

This trace showed show that someone had set up a rule for the Watchlist for 159.226.63.190 which is a China IP. It appeared many times throughout the logs. There is definitely a interest at this China IP in the network. The WhoIs database gave the following:

```
The Computer Network Center Chinese Academy of Sciences (NET-NCFC)
   P.O. Box 2704-10,
   Institute of Computing Technology Chinese Academy of Sciences
   Beijing 100080, China
```

GIAC 000218 VA-CIRT port 34555

```
07/19-14:51:02.951149  [**] GIAC 000218 VA-CIRT port 34555 [**] 163.120.1.212:25 ->
MY.NET.253.24:34555
07/26-07:13:53.374962  [**] GIAC 000218 VA-CIRT port 35555 [**] 204.101.251.57:25 ->
MY.NET.253.24:35555
07/27-08:04:04.570051  [**] GIAC 000218 VA-CIRT port 34555 [**] 129.132.178.196:113 ->
MY.NET.100.230:34555
07/30-07:32:39.903167  [**] GIAC 000218 VA-CIRT port 34555 [**] 152.163.224.100:25 ->
MY.NET.253.24:34555
```

This trace shows that a rule setup to detect activity on port 34555 has fired several times in the logs we reviewed. One point of significance is that the source port is always on a well known assigned port 25 (SMTP), 113 (Ident), 53 (DNS). It would be worth doing a trojan scan to see if indeed there was a trojan on the machines. Trojan Trinoo is known to lurk on UDP port 34555.

Null Scan!

```
07/26-15:24:39.163260  [**] Null scan! [**] 172.138.37.179:15091 -> MY.NET.253.112:443
07/29-02:32:54.058776  [**] Null scan! [**] 208.46.220.122:6688 -> MY.NET.98.166:1055
07/29-03:44:14.119096  [**] Null scan! [**] 24.18.166.130:1963 -> MY.NET.100.236:6346
07/29-06:27:46.015408  [**] Null scan! [**] 62.136.29.13:1418 -> MY.NET.100.236:6346
```

This trace shows a Null Scan! Alert. A null scan is where none of the TCP flags are set. This is not the normal condition and is typically indicative of some sort of OS fingerprinting. A normal TCP connection has at least one flag set. A tool such as nmap can be used for this exploit and typically sends a bunch of anomalous stimuli when attempting to fingerprint the OS and examining the way a particular TCP/IP stack responds.

Below are vulnerabilities listed at the CVE site. (http://cve.mitre.org)

Name	Description
CAN-1999-0454	** CANDIDATE (under review) ** A remote attacker can sometimes identify the operating system of a host based on how it reacts to some IP or ICMP packets, using a tool such as nmap or queso.

NMAP TCP Ping!

```
07/27-02:54:34.936909  [**] NMAP TCP ping! [**] 209.218.228.46:80 -> MY.NET.1.8:53
07/27-02:54:39.888327  [**] NMAP TCP ping! [**] 209.218.228.46:80 -> MY.NET.1.8:53
07/27-02:54:39.888376  [**] NMAP TCP ping! [**] 209.218.228.46:53 -> MY.NET.1.8:53
08/04-08:01:02.191197  [**] NMAP TCP ping! [**] 195.25.86.2:80 -> MY.NET.179.77:80
08/04-11:18:28.348261  [**] NMAP TCP ping! [**] 205.128.11.157:80 -> MY.NET.1.8:53
```

This trace shows a NMAP ping alert to port 53 (DNS) and Port 80 (http). This could be reconnaissance or OS fingerprinting.

Below are vulnerabilities listed at the CVE site. (http://cve.mitre.org)

Name	CAN-1999-0454 (under review)
Description	A remote attacker can sometimes identify the operating system of a host based on how it reacts to some IP or ICMP packets, using a tool such as nmap or queso.

SNMP public access

```
07/26-09:33:38.673441  [**] SNMP public access [**] MY.NET.97.186:1048 -> MY.NET.101.192:161
```

This trace shows an SNMP alert that actually showed up often in the 30 days of scans. It seems that you have a router that is configured to use the 'public' string. This si not a wise thing to do and should be change from this default setting.

Below are vulnerabilities listed at the CVE site. (http://cve.mitre.org)

Name	Description
CVE-1999-0472	The SNMP default community name "public" is not properly removed in NetApps C630 Netcache, even if the administrator tries to disable it.
CAN-1999-0517	** CANDIDATE (under review) ** An SNMP community name is the default (e.g. public), null, or missing.

SUNRPC highport access! and Attempted SUN RPC high port access

```
07/29-16:24:35.903923  [**] SUNRPC highport access! [**] 205.188.3.205:5190 ->
MY.NET.98.145:32771
08/03-11:15:26.051255  [**] Attempted Sun RPC high port access [**] 205.188.153.111:4000 ->
MY.NET.217.126:32771
08/05-02:47:09.846709  [**] SUNRPC highport access! [**] 192.102.249.3:25 ->
MY.NET.130.94:32771
```

This trace shows the SUNRPC highport access alert. This is probably a reconnaissance probe to port 111 (sunrpc).

Below are vulnerabilities listed at the CVE site. (http://cve.mitre.org)

CVE-1999-0320	SunOS rpc.cmsd allows attackers to obtain root access by overwriting arbitrary files.

Scan from possible DNS server

```
Jul 30 06:12:39 MY.NET.1.3:53 -> MY.NET.101.89:65512 UDP
Jul 30 06:12:39 MY.NET.1.3:53 -> MY.NET.101.89:65513 UDP
Jul 30 06:12:39 MY.NET.1.3:53 -> MY.NET.101.89:65514 UDP
Jul 30 06:12:39 MY.NET.1.3:53 -> MY.NET.101.89:65515 UDP
Jul 30 06:12:39 MY.NET.1.3:53 -> MY.NET.101.89:65516 UDP
```

This trace shows a UDP scan from your DNS server to another of your network's IPs. This server needs to be investigated and see if it has been compromised.

Happy 99 Virus

```
08/05-11:22:48.017066  [**] Happy 99 Virus [**] 206.67.51.242:4889 -> MY.NET.6.47:25
```

There was at least one instance of a Happy 99 Virus showing up.

TELNET

```
08/05-18:47:13.982488  [**] IDS127 - TELNET - Login Incorrect [**] MY.NET.60.8:23 ->
207.172.151.22:1674
```

There was an attempt to login on the telnet port.

Below are vulnerabilities listed at the CVE site. (http://cve.mitre.org)

Name	Description
CVE-1999-0073	Telnet allows a remote client to specify environment variables including LD_LIBRARY_PATH, allowing an attacker to bypass the normal system libraries and gain root access.
CVE-1999-0087	Denial of service in AIX telnet can freeze a system and prevent users from accessing the server.
CVE-1999-0192	Buffer overflow in telnet daemon tgetent routing allows remote attackers to gain root access via the TERMCAP environmental variable.
CVE-1999-0230	Buffer overflow in Cisco 7xx routers through the telnet service.

CVE-1999-0273	Denial of service through Solaris 2.5.1 telnet by sending ^D characters.
CVE-1999-0290	The WinGate telnet proxy allows remote attackers to cause a denial of service via a large number of connections to localhost.
CVE-1999-0416	Vulnerability in Cisco 7xx series routers allows a remote attacker to cause a system reload via a TCP connection to the router's TELNET port.
CVE-1999-0740	Remote attackers can cause a denial of service on Linux in.telnetd telnet daemon through a malformed TERM environmental variable.
CVE-1999-0749	Buffer overflow in Microsoft Telnet client in Windows 95 and Windows 98 via a malformed Telnet argument.
CVE-1999-0817	Lynx WWW client allows a remote attacker to specify command-line parameters which Lynx uses when calling external programs to handle certain protocols, e.g. telnet.
CVE-1999-0889	Cisco 675 routers running CBOS allow remote attackers to establish telnet sessions if an exec or superuser password has not been set.
CVE-1999-0991	Buffer overflow in GoodTech Telnet Server NT allows remote users to cause a denial of service via a long login name.
CVE-2000-0113	The SyGate Remote Management program does not properly restrict access to its administration service, which allows remote attackers to cause a denial of service, or access network traffic statistics.
CVE-2000-0152	Remote attackers can cause a denial of service in Novell BorderManager 3.5 by pressing the enter key in a telnet connection to port 2000.
CVE-2000-0212	InterAccess TelnetID Server 4.0 allows remote attackers to conduct a denial of service via malformed terminal client configuration information.
CVE-2000-0268	Cisco IOS 11.x and 12.x allows remote attackers to cause a denial of service by sending the ENVIRON option to the Telnet daemon before it is ready to accept it, which causes the system to reboot.
CAN-1999-0285	** CANDIDATE (under review) ** Denial of service in telnet from the Windows NT Resource Kit, by opening then immediately closing a connection.
CAN-1999-0571	** CANDIDATE (under review) ** A router allows arbitrary hosts to connect to its configuration service, or related services such as telnet.
CAN-1999-0619	** CANDIDATE (under review) ** The Telnet service is running.
CAN-1999-0843	** CANDIDATE (under review) ** Denial of service in Cisco routers running NAT via a PORT command from an FTP client to a Telnet port.
CAN-1999-0919	** CANDIDATE (under review) ** A memory leak in a Motorola CableRouter allows remote attackers to conduct a denial of service via a large number of telnet connections.
CAN-2000-0166	** CANDIDATE (under review) ** Buffer overflow in the InterAccess telnet server TelnetD allows remote attackers to execute commands via a long login name.

CAN-2000-0480	** CANDIDATE (under review) ** Dragon telnet server allows remote attackers to cause a denial of service via a long username.
CAN-2000-0581	** CANDIDATE (under review) ** Windows 2000 Telnet Server allows remote attackers to cause a denial of service by sending a continuous stream of binary zeros, which causes the server to crash.
CAN-2000-0598	** CANDIDATE (under review) ** Fortech Proxy+ allows remote attackers to bypass access restrictions for to the administration service by redirecting their connections through the telnet proxy.
CAN-2000-0665	** CANDIDATE (under review) ** AMSoft TelSrv telnet server 1.5 and earlier allows remote attackers to cause a denial of service via a long username.

Napster 8888 Data and Napster 7777 Data

```
08/05-18:30:10.125959  [**] Napster 8888 Data [**] 208.184.216.191:8888 -> MY.NET.201.2:1463
08/05-18:30:45.334122  [**] Napster 8888 Data [**] 208.184.216.191:8888 -> MY.NET.201.2:1463
08/05-18:30:45.334201  [**] Napster 8888 Data [**] MY.NET.201.2:1463 -> 208.184.216.191:8888
08/05-18:35:16.695439  [**] Napster Client Data [**] 148.231.51.2:2114 -> MY.NET.97.229:6699
08/05-18:45:31.071771  [**] Napster 7777 Data [**] 208.184.216.183:7777 ->
MY.NET.97.204:3419
08/05-18:45:31.072253  [**] Napster 7777 Data [**] 208.184.216.183:7777 ->
MY.NET.97.204:3419
08/05-18:45:31.100537  [**] Napster 7777 Data [**] 208.184.216.183:7777 ->
MY.NET.97.204:3419

This trace shows that the Napster alert went off because there is Napster traffic. These
machines
should be investigated.

Below are vulnerabilities listed at the CVE site. (http://cve.mitre.org)
```

Name	Description
CAN-2000-0281	** CANDIDATE (under review) ** Buffer overflow in the Napster client beta 5 allows remote attackers to cause a denial of service via a long message.
CAN-2000-0412	** CANDIDATE (under review) ** The gnapster and knapster clients for Napster do not properly restrict access only to MP3 files, which allows remote attackers to read arbitrary files from the client by specifying the full pathname for the file.

Conclusion

These traces show that there is a lot of possibly hostile traffic hitting your network. Since this is a preliminary analysis
it is suggested that a round the clock scan be put into place and monitored by an analyst and that the rulesbase is refined.
This would be a good time to re-examine your perimeter defenses and make sure that the traffic that was alerted in the Snort

sensor is being addressed at the firewall level.

Assignment 4 – back to the top

Analysis process: please describe the process you used to analyze the data in assignment 3.

First let me say that other than what I saw and learned at the conference in Ottawa about Snort that this is the extent of my knowledge of the tool. It has just been in the last month that I loaded WinDump to try and do some analyzing to prepare for this practical. I did not set up any rules, so all I had was the full log which took many hours of review to pull out things that looked questionable. I am from a Windows NT background and am just now beginning to plan to use Linux to install a IDS or combination IDS of Shadow or Snort. I have been told that this is a monumental effort for one that has the background I have.

As for these logs I looked at everyone of them in WordPad and pulled what was questionable out. I then used notes from Ottawa, the CVE site, the Sans site and a book by Stephen Northcutt, Network Intrustion Detection and Stevens TCP/IP Volume 1.

Last Updated: September 21st, 2017

Upcoming Training

SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS October Singapore 2017	Singapore, Singapore	Oct 09, 2017 - Oct 28, 2017	Live Event
Community SANS Ottawa SEC503	Ottawa, ON	Oct 16, 2017 - Oct 21, 2017	Community SANS
SANS Berlin 2017	Berlin, Germany	Oct 23, 2017 - Oct 28, 2017	Live Event
SANS San Diego 2017	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	Live Event
San Diego Fall 2017 - SEC503: Intrusion Detection In-Depth	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	vLive
SANS Seattle 2017	Seattle, WA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Paris November 2017	Paris, France	Nov 13, 2017 - Nov 18, 2017	Live Event
Community SANS Pensacola SEC503	Pensacola, FL	Nov 27, 2017 - Dec 02, 2017	Community SANS
SIEM & Tactical Analytics Summit & Training	Scottsdale, AZ	Nov 28, 2017 - Dec 05, 2017	Live Event
SANS Munich December 2017	Munich, Germany	Dec 04, 2017 - Dec 09, 2017	Live Event
SANS Cyber Defense Initiative 2017	Washington, DC	Dec 12, 2017 - Dec 19, 2017	Live Event
SANS Security East 2018	New Orleans, LA	Jan 08, 2018 - Jan 13, 2018	Live Event
SANS Las Vegas 2018	Las Vegas, NV	Jan 28, 2018 - Feb 02, 2018	Live Event
SANS London February 2018	London, United Kingdom	Feb 05, 2018 - Feb 10, 2018	Live Event
SANS Dallas 2018	Dallas, TX	Feb 19, 2018 - Feb 24, 2018	Live Event
SANS Northern VA Spring - Tysons 2018	Tysons, VA	Mar 17, 2018 - Mar 24, 2018	Live Event
SANS Secure Canberra 2018	Canberra, Australia	Mar 19, 2018 - Mar 24, 2018	Live Event
SANS 2018	Orlando, FL	Apr 03, 2018 - Apr 10, 2018	Live Event
SANS Baltimore Spring 2018	Baltimore, MD	Apr 21, 2018 - Apr 28, 2018	Live Event
SANS OnDemand	Online	Anytime	Self Paced
SANS SelfStudy	Books & MP3s Only	Anytime	Self Paced